

# 基于ResNet-50迁移学习的植物纤维图像分类系统

项目汇报

植物纤维识别项目组

2025 年 6 月 5 日

# 汇报大纲

① 项目概述

② 技术方案

③ 实验结果

④ 项目成果

⑤ 总结与展望

# 项目背景与目标

- **研究背景：**植物纤维识别在材料科学、生物学研究具有重要意义
- **项目目标：**探索基于深度学习的植物纤维自动分类可行性
- **技术路线：**采用ResNet-50预训练模型进行迁移学习
- **当前阶段：**初步验证技术方案，识别关键挑战

## 探索重点

- 多模态图像融合（光镜+电镜）
- 小样本数据集的处理策略
- 类别不平衡问题的解决方案

# 数据集概况

## 数据集统计:

- 总图像数量: 295张
- 光镜图像: 147张 (49.8%)
- 电镜图像: 148张 (50.2%)
- 植物类别: 6种

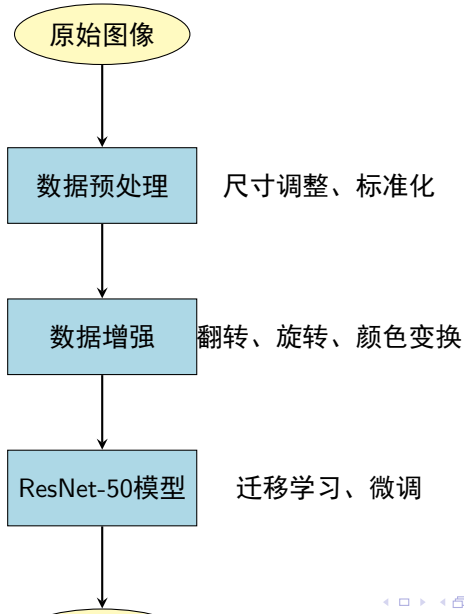
## 图像特征:

- 主要尺寸: 1600×1200像素
- 文件格式: TIFF、PNG
- 平均大小: 3.78MB

## 类别分布:

植物类别	图像数量
粽叶芦	65
怀槐	54
硬头黄	52
长叶水麻	49
鼠皮树	42
蒙古栎	33

# 系统架构设计



## 网络结构:

- **骨干网络:** ResNet-50 (ImageNet预训练)
- **特征提取:** 2048维特征向量
- **分类头:** 全连接层 + Dropout
- **输出层:** 6类softmax分类

## 迁移学习策略:

- 冻结前期卷积层参数
- 微调后期特征层
- 重新训练分类头

## 模型参数

- 输入尺寸:  $224 \times 224 \times 3$
- 批次大小: 32
- 学习率: 0.001
- 优化器: Adam
- 训练轮数: 50
- 权重衰减:  $1e-4$

## 为什么选择ResNet-50?

- 传统深度网络存在“梯度消失”问题：信息在反向传播时逐层衰减
- ResNet通过“跳跃连接”让信息可以直接传递到后面的层
- 就像给信息建了一条“高速公路”，避免拥堵

## 残差块的核心思想：

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + \mathbf{x}$$

输出 = 学到的变化 + 原始输入（这样即使学不到什么，至少能保持原样）

## Bottleneck结构（降低计算量）：

$1 \times 1$ 卷积 → 降维（减少通道数）

$3 \times 3$ 卷积 → 特征提取（核心计算）

$1 \times 1$ 卷积 → 升维（恢复通道数）

# 损失函数与优化策略

## 如何衡量模型的“错误程度”？

- 使用交叉熵损失：预测越准确，损失越小
- 当模型完全猜对时，损失为0；完全猜错时，损失很大

## 交叉熵损失函数：

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C y_{ij} \log(\hat{y}_{ij})$$

$N$ =样本数量， $C$ =类别数量， $y_{ij}$ =真实标签， $\hat{y}_{ij}$ =预测概率

**Softmax：将输出转换为概率**

$$\hat{y}_{ij} = \frac{\exp(z_{ij})}{\sum_{k=1}^C \exp(z_{ik})}$$

确保所有类别的预测概率加起来等于1

**Adam优化器：智能调整学习步长**

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{\hat{\mathbf{v}}_t} + \epsilon} \hat{\mathbf{m}}_t$$

结合了动量和自适应学习率，训练更稳定



## 数据预处理:

- 图像尺寸标准化:  $224 \times 224$
- RGB格式转换
- ImageNet标准化
- 多格式支持 (TIFF/PNG/JPG)

## 数据增强:

- 随机水平/垂直翻转
- 随机旋转 ( $\pm 30^\circ$ )
- 颜色抖动变换
- 随机裁剪

## 数据集划分:

- 训练集: 70%
- 验证集: 15%
- 测试集: 15%
- 分层采样保证类别平衡

## 不平衡处理:

- 加权随机采样
- 类别权重调整
- 数据增强补偿

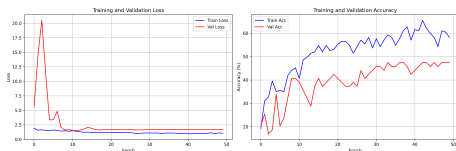


图: 训练历史曲线

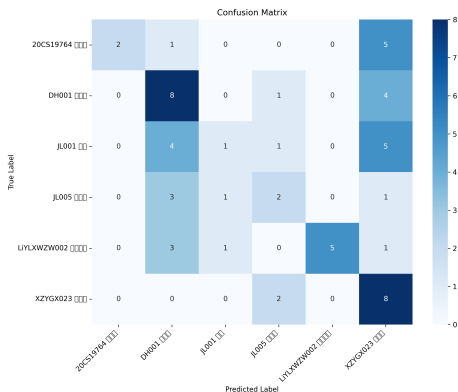
## 训练观察:

- 训练损失下降较快
- 验证准确率提升有限
- 存在训练验证差距
- 收敛到局部最优

## 关键指标

- 最佳验证准确率: 47.46%
- 测试集准确率: 44.07%
- 训练时间: 约33分钟

# 分类性能评估



图：混淆矩阵

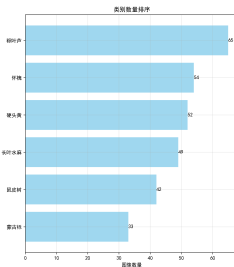
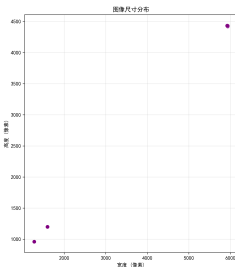
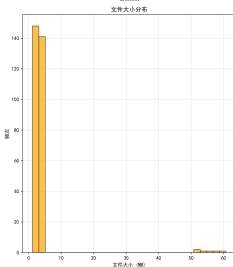
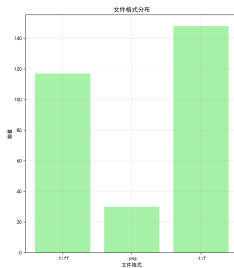
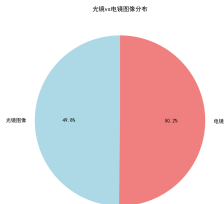
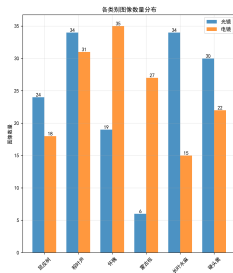
## 性能分析（存在明显问题）：

- 整体准确率偏低(44.07%)，远未达到实用标准
- 各类别表现差异很大，模型学习不充分
- 数据量不足是主要瓶颈(仅295张图像)
- 类别不平衡加剧了分类困难

## 评估指标

- 宏平均精确率：57%
- 宏平均召回率：42%
- 宏平均F1分数：41%
- 加权平均准确率：44.07%

# 数据集分析可视化



图：数据集统计分析图表

# 项目交付物

## 核心代码模块:

- train.py - 模型训练脚本
- predict.py - 预测推理脚本
- analyze\_dataset.py - 数据分析工具

## 配置文件:

- config.json - 训练参数配置
- requirements.txt - 依赖包列表
- .gitignore - 版本控制配置

## 模型文件:

- 训练好的ResNet-50模型
- 模型权重和配置信息
- 类别映射关系

## 分析报告:

- 数据集统计报告 (JSON)
- 训练历史可视化
- 混淆矩阵分析
- 性能评估报告

## 技术亮点

- **多模态融合**: 统一处理光镜和电镜图像, 提升模型泛化能力
- **迁移学习**: 充分利用ImageNet预训练权重, 加速收敛
- **数据增强**: 丰富的增强策略, 提高模型鲁棒性
- **不平衡处理**: 加权采样策略, 解决类别不均衡问题

## 工程实践

- **模块化设计**: 代码结构清晰, 易于维护和扩展
- **完整文档**: 详细的README和代码注释
- **可视化分析**: 丰富的图表和统计信息
- **自动化流程**: 一键训练和预测功能

# 项目总结

## 已完成工作

- ☐ 完成数据集收集和预处理
- ☐ 实现基于ResNet-50的分类模型
- ☐ 完成模型训练和性能评估
- ☐ 开发预测和分析工具
- ☐ 生成完整的项目文档

## 阶段性成果

- 完成了技术方案的初步验证
- 识别了数据不足等关键瓶颈
- 建立了基础的分类流水线
- 为后续改进提供了基准

## 当前挑战

- 准确率44.07%距离实用化还有很大差距
- 数据集规模严重不足

# 下一步改进计划

## 紧急优先级 - 数据问题

- 扩大数据集：目标收集至少1000张图像
- 平衡类别分布：确保每类至少100张样本
- 提高数据质量：标注一致性检查

## 中期目标 - 技术优化

- 尝试更适合小样本的网络架构
- 改进数据增强策略
- 探索半监督学习方法
- 引入领域自适应技术

## 长期规划 - 系统完善

- 达到80%以上的实用准确率
- 开发用户友好的应用界面
- 建立持续学习机制